

Conectando Aplicações em Visual Basic a Base de Dados MySQL

A. Pré-Requisitos (instalar):

- Visual Studio 2015 (ou versão anterior)
- **XAMPP** ou WAMPP ou MySQL Workbench
- MySQL Connector Net 6.2.3 (ou versão anterior)

B. Base de Dados - No XAMPP ou qualquer outro sistema indicado acima

Nome da base de dados: **sgedb**

Tabela - Nesta primeira parte iremos simplesmente trabalhar com a tabela **usuario**:

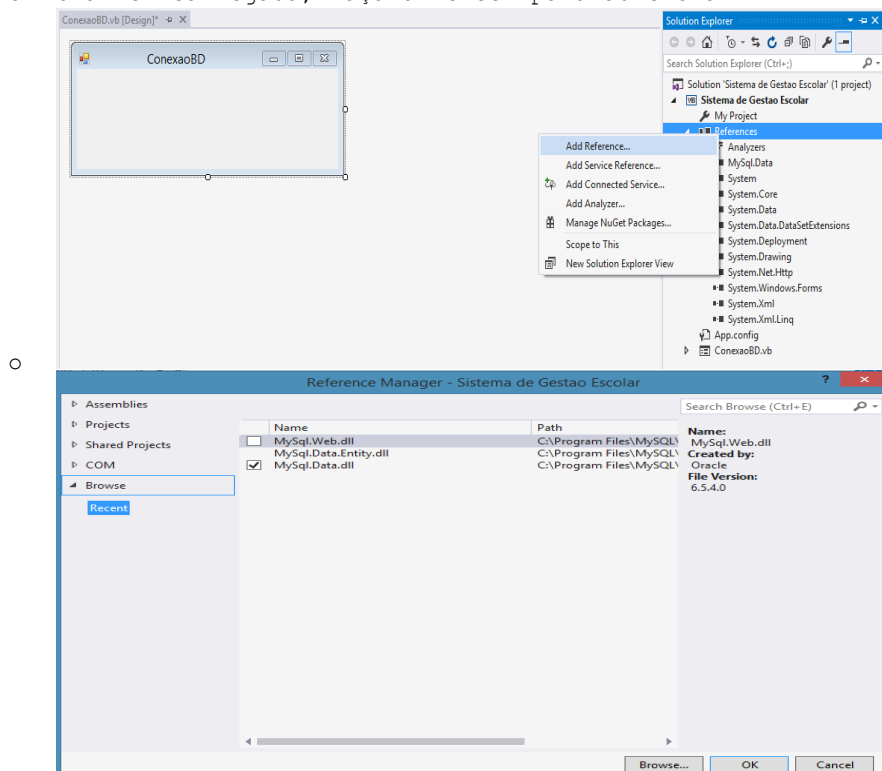
- **usuario**(id, nome_usuario, palavra_passe, nome);
- **classe**(id_classe, designacao);
- **turma**(id_turma, designacao);
- **avaliacao**(id_avaliacao, teste1, teste2, trabalho1, media, classificacao);
- **estudante**(id_estudante, nome, sexo, idade, id_classe, id_turma, id_avaliacao);

Código para criação da tabela **usuário** - Vide o Anexo A

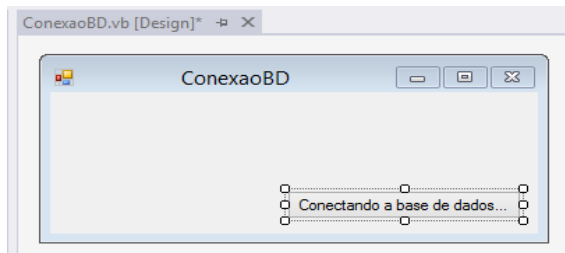
C. Aplicação - No Visual Studio

Nome da Solução/do projecto: Sistema de Gestão Escolar

- Renomear o formulário **Form1.vb** para **ConexaoBD.vb**
- Adicionar a referência **MySQL.Data** à solução ou ao projecto
 - Sobre o **Solution Explorer**, clique com o lado direito do mouse sobre **References** -> **Add Reference**
 - No **Reference Manager** -> **Browse**
 - Pesquise **C:\Program Files\MySQL\Connector NET 6.5.4\Assemblies\v4.0** e seleccione **MySQL.Data.dll**
 - Uma vez carregada, faça um check para adicionar



No formulário **ConexaoBD** (modo de design), faça double click sobre o botão **Conectando a base de dados...** para adicionar o evento (código de conexão com a bd)



No modo **View Code**, importar o Namespace **MySQL.Data.MySqlClient** (linha 1) para que haja acesso as várias classes de manipulação a base de dados

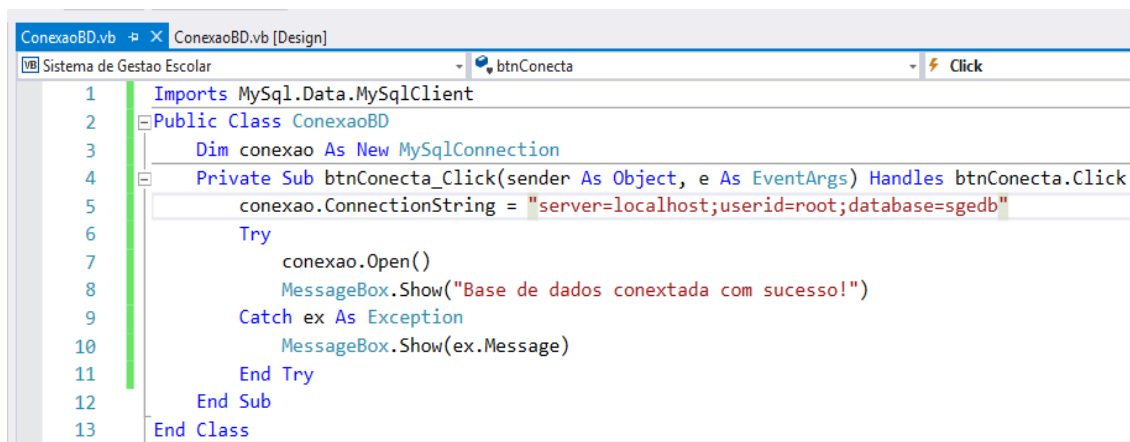
Criar uma instância da classe **MySQLConnection** através da variável **conexao** (linha 3) por forma que se crie a conexão.

Estabelecer a conexão através da propriedade **ConnectionString**, indicando sequencialmente o servidor (**server**), o id do usuário (**userid**), a palavra-passe (**password**) e a base de dados (**database**). De acordo com o nosso exemplo a propriedade **ConnectionString** da nossa classe **conexao** recebe os seguintes valores: (linha 5)

Nota: No nosso exemplo não consta o atributo **password** uma vez que ela não está definido. Escrever **"password="** também indica a senha como sendo nula.

Abrir a conexão através do método **Open()** e exibir uma mensagem de sucesso (linhas 7 e 8, respectivamente)

Exibir uma mensagem de erro com a respectiva causa (linha 10), caso ocorra.

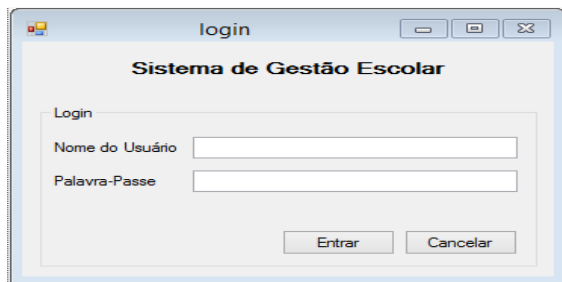


Formulário de Login

No XAMPP - Insira os seguintes dados na base de dados **sgedb**

- nome_usuario - seu usuário (Ex. amarenjua)
- palavra_passe - sua senha (Ex. alberto)
- nome - seu nome completo (Ex. Alberto Maregue Marenjua)

No visual Studio - Crie o formulário **login**



Faça double click sobre o Botão **Cancelar** e insira o código para encerrar a aplicação invocando o método **Exit()**-sair, à classe **Application**-a nossa aplicação (linha 9)

No modo de **View Code**, importa o Namespace **MySQL.Data.MySqlClient** (linha 1)

Criar uma variável **conexao** do tipo **MySQLConnection** (linha 3); variável **comando** do tipo **MySQLCommand** (linha 4) - para executar a consulta (query) na base de dados; variável **ler** do tipo **MySQLDataReader** (linha 5) - para ler os dados da base de dados; variável **consulta** do tipo **String** (linha 6) - para criar as consultas, no caso vertente, para selecionar todos os dados da base de dados de acordo com a condição estabelecida; e a variável **contador** do tipo **Integer** (linha 7) - para controlar o número de registos selecionados na base de dados.

```
login.vb* - X login.vb [Design] ConexaoBD.vb ConexaoBD.vb [Design]
Sistema de Gestao Escolar btnEntrar Click
1 Imports MySQL.Data.MySqlClient
2 Public Class login
3     Dim conexao As MySqlConnection
4     Dim comando As MySQLCommand
5     Dim ler As MySQLDataReader
6     Dim consulta As String
7     Dim contador As Integer
8 Private Sub btnSair_Click(sender As Object, e As EventArgs) Handles btnSair.Click
9     Application.Exit()
10 End Sub
11
```

No modo de **design**, clique por duas vezes sobre o Botão **Entrar**. Dentro do espaço para introduzir o evento (bloco de instruções), na linha 13, verifica-se se os campos **txt.Usuario** e **txt.Senha** não estão vazios. Caso seja falso, uma mensagem de erro é emitida pelo sistema (linha 36). Se a condição for verdadeira (os campos não forem iguais a null), é criada uma instância da classe **MySQLConnection** linha (14) e se estabelece uma conexão linha (15) através da propriedade **ConnectionString**, indicando sequencialmente o servidor (**server**), o id do usuário (**userid**), a palavra-passe (**password**) e a base de dados (**database**).

Cria-se uma instrução **Try** (linha 16) que consiste em bloco **Try** que contém várias declarações e uma cláusula **Catch** (linha 31) em caso da tentativa falhar. Constituem declarações do bloco **Try** a abertura da conexão a través do método **Open()** - linha 17; a criação da consulta na base de dados (linha 18) - selecionar todos os dados na tabela **usuario** em que **nome_usuario** e **palavra_passe** são iguais as introduzidas pelo usuário; criação de uma instância da classe **MySQLCommand** (linha 19) que recebe como parâmetro a consulta e a conexão a base de dados; Execução da consulta e atribuição do resultado a variável **ler** do tipo **MySQLDataReader** (linha 20).

Enquanto existirem registos na base de dados referente a consulta em causa (verificado através da propriedade **ler.Read** - linha 22), incrementa-se mais uma unidade ao valor da variável **contador** (linha 23).

Se o contador for igual 1 (linha 25), chama-se, através do método **Show()**, o formulário **Main** (a ser criado a posterior - formulário principal) e esconde-se, através do método **Hide()**, o actual formulário **Me** (formulário de login).

```

login.vb [Design]  ConexaoBD.vb  ConexaoBD.vb [Design]
Sistema de Gestao Escolar  btnEntrar  Click
12 Private Sub btnEntrar_Click(sender As Object, e As EventArgs) Handles btnEntrar.Click
13     If txtUsuario.Text <> "" Or txtSenha.Text <> "" Then
14         conexao = New MySqlConnection
15         conexao.ConnectionString = "server=localhost;userid=root;password=;database=sgebd"
16         Try
17             conexao.Open()
18             consulta = "select * from usuario where nome_usuario='" + txtUsuario.Text + "' AND palavra_passe"
19             comando = New MySqlCommand(consulta, conexao)
20             ler = comando.ExecuteReader
21             contador = 0
22             While ler.Read
23                 contador = contador + 1
24             End While
25             If contador = 1 Then
26                 Main.Show()
27                 Me.Hide()
28             ElseIf contador < 1 Then
29                 MessageBox.Show("Usuário ou Palavra-passe incorretos!", "Erro", MessageBoxButtons.OK, Message
30             End If
31         Catch ex As MySqlException
32             MessageBox.Show(ex.Message)
33         End Try
34     Else
35         MessageBox.Show("Campos vazios", "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error)
36     End If
37 End Sub

```

Formulário Principal (Main)

Cria um novo formulário (nome - Main) e coloca-o como **MDI** atribuindo o valor **True** a sua propriedade **IsMdiContainer**. Adicione ao Main um menu através da ferramenta **MenuStrip** - usando o painel **ToolBox** e seu submenus, conforme ilustram as imagens abaixo.



Navegue sobre **Ficheiro** -> **Fechar** -> e faça double click para adicionar o evento que consiste em fechar a aplicação através da classe e propriedade **Application.Exit**.

No modo de design, navegue sobre **Ficheiro** -> **Novo** -> **Usuário** -> e faça double clique para adicionar um evento. Na linha 7, cria uma instância do objecto **CriarUsuario** (criar formulário); defina a instância criada como sendo um formulário filho do **MDI** Main através da propriedade **MdiParent** (linha 8); apresente o formulário **CriarUsuario** através do método **Show()** linha 9.

No modo de design, navegue sobre **Ver** -> **Usuário** -> e faça double clique para adicionar um evento (linhas 13-15). **Nota:** criar formulário **VerUsuario**

```

Main.vb [Design]  login.vb  login.vb [Design]  ConexaoBD.vb  ConexaoBD.vb [Design]
Sistema de Gestao Escolar  Main  InitializeComponent
1 Public Class Main
2     Private Sub menuItemFechar_Click(sender As Object, e As EventArgs) Handles menuItemFechar.Click
3         Application.Exit()
4     End Sub
5
6     Private Sub mnNovoUsuario_Click(sender As Object, e As EventArgs) Handles mnNovoUsuario.Click
7         Dim cn As New CriarUsuario
8         cn.MdiParent = Me
9         cn.Show()
10    End Sub
11
12    Private Sub mnVerUsuario_Click(sender As Object, e As EventArgs) Handles mnVerUsuario.Click
13        Dim vn As New VerUsuario
14        vn.MdiParent = Me
15        vn.Show()
16    End Sub
17 End Class

```

Formulário para adicionar usuários ao SGE (CriarUsuario)

Ao apresentar o formulário na tela, serão apresentados os últimos cinco usuários registados no sistema. Para o efeito, no **ToolBox**, pesquise por **DataGrid** e insira-o dentro do **GroupBox** "Lista dos últimos usuários adicionados ao sistema". Desactive todas as opções de inserção, actualização e remoção de dados através da tabela.

Para adicionar um icon a um Botão, selecione o Botão pretendido -> pesquise pela propriedade **Image** -> escolha o icon correspondente a acção do Botão. Pode definir a posição do icon em relação ao texto através da propriedade **TextImageRelation**.

A função do Botão **Limpar** consiste em limpar qualquer campo de texto no formulário. Para adicionar o evento correspondente, faça double click sobre o Botão **Limpar** e chame o método **limparCampos()** linha 8 - a ser criado já a seguir. Imediatamente abaixo, crie o método **limparCampos()** e adicione o bloco de código conforme na imagem que se segue. Depois de limpar os campos através do método **Clear()** - linha 12-14, devolver-se-á o foco ao campo de texto txtNome através do método **Focus()** - linha 15.

```

7 Private Sub btnLimpar_Click(sender As Object, e As EventArgs) Handles btnLimpar.Click
8     limparCampos()
9 End Sub
10
11 Public Sub limparCampos()
12     txtNome.Clear()
13     txtUsuario.Clear()
14     txtSenha.Clear()
15     txtNome.Focus()
16 End Sub

```

Apresentação de dados da base de dados ao DataGrid do formulário

No modo de code, nas linhas 1, 3-6, adicione as instruções conforme a imagem.

```

1 Imports MySql.Data.MySqlClient
2 Public Class CriarUsuario
3     Dim conexao As MySqlConnection
4     Dim comando As MySqlCommand
5     Dim ler As MySqlDataReader
6     Dim query As String

```

No modo de Design, faça double click sobre o formulário (Ex. no título) para inserir o evento. Na linha 51, chame pelo método **leUltimosUsuariosAdicionados()** - a ser criado já a seguir.

Imediatamente abaixo, crie o método **leUltimosUsuariosAdicionados()**, conforme a imagem que se segue.

```

50 Private Sub CriarUsuario_Load(sender As Object, e As EventArgs) Handles MyBase.Load
51     leUltimosUsuariosAdicionados()
52 End Sub
53
54 Public Sub leUltimosUsuariosAdicionados()
55     conexao = New MySqlConnection
56     conexao.ConnectionString = "server=localhost;userid=root;password=;database=sgedb"
57     Dim adaptador As New MySqlDataAdapter
58     Dim tabela As New DataTable
59     Dim bsource As New BindingSource
60     Try
61         conexao.Open()
62         query = "select * from usuario order by id desc limit 5"
63         comando = New MySqlCommand(query, conexao)
64         adaptador.SelectCommand = comando
65         adaptador.Fill(tabela)
66         bsource.DataSource = tabela
67         DataGridView1.DataSource = bsource
68         adaptador.Update(tabela)
69         conexao.Close()
70     Catch ex As MySqlException
71         MessageBox.Show(ex.Message)
72     End Try
73 End Sub

```

Depois de se estabelecer e abrir a conexão com a base de dados, a feita a consulta (selecionar todos os campos da tabela **usuario**) ordenando e limitando os registos, isto é, serão apresentados 5 usuários ordenados pelo ID do maior ao menor (se existirem) - linha 62. Executa-se a consulta e o resultado é apresentado em tabela no **DataGrid** criado - linhas 63-67.

Salvar usuários na base de dados

No modo de design, faça double click sobre o Botão **Gravar** para adicionar o evento correspondente.

Depois de verificar se os campos estão vazios, é estabelecida e aberta a conexão com a base de dados, linhas 22,23,25. A posterior, insere-se na tabela **usuario** os valores dos campos de texto do usuário, da senha e do nome, conforme descrito na **query**, linha 26. Executa-se a **query**, fecha-se a conexão e chama-se pelo método **leUltimosUsuariosAdicionados()** - linha 31 - e pelo método **limparCampos()** -linha 32- para actualizar os dados no **DataGrid** e limpar os campos de texto, respectivamente.

Formulário para Editar/Remover usuários (VerUsuario)

No modo de code, entre as linhas 4-7 foram declaradas variáveis de uso já conhecidos, sendo que na linha 8 a variável **ID** armazenará o id do usuário a cada momento que este for selecionado na **ComboBox**.

```

1 Imports MySql.Data.MySqlClient
2
3 Public Class VerUsuario
4     Dim conexao As MySqlConnection
5     Dim comando As MySqlCommand
6     Dim ler As MySqlDataReader
7     Dim query As String
8     Dim ID As Integer = 0

```

Ao carregar no formulário, pretende-se que o sistema preencha a **DataGrid** e a **ComboBox** com os elementos da base de dados. Para o efeito, faça double click sobre o formulário para adicionar o evento. Chame a função **leUsuarios()** - linha 11, a ser criada a seguir.

```

10 Private Sub VerUsuario_Load(sender As Object, e As EventArgs) Handles MyBase.Load
11     leUsuarios()
12 End Sub

```

Imediatamente abaixo, crie o método **leUsuarios()**. Estabeleça a **conexão** com a base de dados, linhas 15 e 16, selecione todos os campos da tabela **usuario** - linha 19 - e adicione como item da **ComboBox** os elementos da coluna nome da tabela **usuário** - linha 25.

```

14 Public Sub leUsuarios()
15     conexao = New MySqlConnection
16     conexao.ConnectionString = "server=localhost;userid=root;password=;database=sgedb"
17     Try
18         conexao.Open()
19         query = "select * from usuario order by nome"
20         comando = New MySqlCommand(query, conexao)
21         ler = comando.ExecuteReader
22         cbUsuario.Items.Clear()
23         cbUsuario.Items.Add("")
24         While ler.Read
25             cbUsuario.Items.Add(ler.GetString("nome"))
26         End While
27         conexao.Close()
28     Catch ex As MySqlException
29         MessageBox.Show(ex.Message)
30     End Try
31     Try
32         conexao.Open()
33         Dim adaptador As New MySqlDataAdapter
34         Dim tabela As New DataTable
35         Dim bsource As New BindingSource
36         query = "select * from usuario"
37         comando = New MySqlCommand(query, conexao)
38         adaptador.SelectCommand = comando
39         adaptador.Fill(tabela)
40         bsource.DataSource = tabela
41         DataGridView1.DataSource = bsource
42         conexao.Close()
43     Catch ex As MySqlException
44         MessageBox.Show(ex.Message)
45     End Try
46 End Sub

```

Para preencher o **DataGrid** use as mesmas linhas de código usadas no formulário **CriarUsuario** ou veja na imagem acima.

Para preencher os campos de texto com os valores correspondente ao usuário selecionado na **ComboBox**, no modo de design, faça double click sobre o mesmo.

Se o valor selecionado for igual a nada (null), os campos de texto serão limpos, linhas 56 e 57, caso contrário, estabelece-se a conexão - linhas 58-60, seleciona-se todos os campos da tabela **usuario** onde o nome seja igual ao valor selecionado na **ComboBox** - linha 63. Preenche-se os campos de texto com cada valor correspondente da base de dados - linhas 67-69 e atribui-se a variável **ID** o **id** do usuário correspondente.

```

55 Private Sub cbUsuario_SelectedIndexChanged(sender As Object, e As EventArgs) Handles cbUsuario.SelectedIndexChanged
56     If (cbUsuario.Text.Equals("")) Then
57         limparCampos()
58     Else
59         conexao = New MySqlConnection
60         conexao.ConnectionString = "server=localhost;userid=root;password=;database=sgedb"
61         Try
62             conexao.Open()
63             query = "select * from usuario where nome = '" + cbUsuario.Text + "'"
64             comando = New MySqlCommand(query, conexao)
65             ler = comando.ExecuteReader
66             While ler.Read
67                 txtNome.Text = ler.GetString("nome")
68                 txtUsuario.Text = ler.GetString("nome_usuario")
69                 txtSenha.Text = ler.GetString("palavra_passe")
70                 ID = ler.GetString("id")
71             End While
72             conexao.Close()
73         Catch ex As MySqlException
74             MessageBox.Show(ex.Message)
75         End Try
76     End If
77
78 End Sub

```

Para actualizar os dados, no modo de design, faça double click sobre o Botão **Actualizar** para adicionar o evento. Estabeleça e abra a conexão - linhas 88,89 e 91, a actualize os dados da tabela **usuario** onde o **id** seja igual a do usuário em causa - linha 92. Lembre-se que a variável **ID** armazena um novo valor a cada momento que é seleccionado um usuário diferente na **ComboBox**.

```

84 Private Sub btnActualizar_Click(sender As Object, e As EventArgs) Handles btnActualizar.Click
85     If txtNome.Text <> "" Then
86         If txtUsuario.Text <> "" Then
87             If txtSenha.Text <> "" Then
88                 conexao = New MySqlConnection
89                 conexao.ConnectionString = "server=localhost;userid=root;password=;database=sgedb"
90                 Try
91                     conexao.Open()
92                     query = "update usuario set nome_usuario = '" + txtUsuario.Text + "', palavra_passe = '" + txtSen
93                     comando = New MySqlCommand(query, conexao)
94                     ler = comando.ExecuteReader
95                     MessageBox.Show("Dados actualizados com sucesso!")
96                     conexao.Close()
97                     leUsuarios()
98                     limparCampos()
99                 Catch ex As MySqlException
100                     MessageBox.Show(ex.Message)
101                 End Try
102             Else
103                 mensagemDeErro()
104             End If
105         Else
106             mensagemDeErro()
107         End If
108     Else
109         mensagemDeErro()
110     End If
111 End Sub

```

Para remover, siga as instruções da imagem.


```

117 Private Sub btnRemover_Click(sender As Object, e As EventArgs) Handles btnRemover.Click
118     If txtNome.Text <> "" Then
119         If txtUsuario.Text <> "" Then
120             If txtSenha.Text <> "" Then
121                 conexao = New MySqlConnection
122                 conexao.ConnectionString = "server=localhost;userid=root;password=;database=sgedb"
123                 Try
124                     conexao.Open()
125                     query = "delete from usuario where id = '" & ID & "'"
126                     comando = New MySqlCommand(query, conexao)
127                     ler = comando.ExecuteReader
128                     MessageBox.Show("Usuário removido com sucesso!")
129                     conexao.Close()
130                     leUsuarios()
131                     limparCampos()
132                 Catch ex As MySqlException
133                     MessageBox.Show(ex.Message)
134                 End Try
135             Else
136                 mensagemDeErro()
137             End If
138         Else
139             mensagemDeErro()
140         End If
141     Else
142         mensagemDeErro()
143     End If
144 End Sub
145 End Class

```

APÊNDICE A

Criar a tabela usuario

```
CREATE TABLE `usuario` (  
  `id` int(10) NOT NULL AUTO_INCREMENT,  
  `nome_usuario` varchar(20) NOT NULL,  
  `palavra_passe` varchar(20) NOT NULL,  
  `nome` varchar(50) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `id_UNIQUE` (`id`),  
  UNIQUE KEY `nome_usuario_UNIQUE` (`nome_usuario`),  
  UNIQUE KEY `palavra_passe_UNIQUE` (`palavra_passe`),  
  UNIQUE KEY `nome_UNIQUE` (`nome`)  
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8 COMMENT='Armazena os usuarios  
do sistema'
```

Inserir dados na tabela usuario

```
INSERT INTO `sgedb`.`usuario`  
(  
  `nome_usuario`,  
  `palavra_passe`,  
  `nome`)  
VALUES  
(  
  "amarenjua",  
  "alberto",  
  "Alberto Marenjua"  
);
```